Crater Detection for Autonomous Landing on Asteroids

Bertrand Leroy, Gérard Medioni
Institute for Robotics and Intelligent Systems
University of Southern California
e-mail:{bleroy,medioni}@iris.usc.edu

Andrew Edie Johnson, Larry Matthies
Jet Propulsion Laboratory
California Institute of Technology
e-mail:{aej,lhm}@robotics.jpl.nasa.gov

Abstract

We describe a visual positioning system for use by a spacecraft to choose a landing site, while orbiting an asteroid. The spacecraft pose is refined using landmarks, such as craters, observed by a visual sensor. The craters, which have an elliptical shape, are detected using a multi-scale method based on voting, and tensors as a representation. We propose a new robust method to infer curvature estimation from noisy sparse data. This method is applied on edge images in order to obtain the oriented normals of the edge curves. Using this information, a dense saliency map corresponding to the position and shape of the craters is computed.The detected craters in the image are matched with the craters projected from a 3-D model, and the best transformation between these two sets is obtained. This system has been tested with both real images of Phobos and a synthetic model.

1 Introduction

Future NASA missions to explore asteroids (and comets) will involve autonomous spacecraft, and should include not only an orbiting phase, but also a landing phase to gather actual samples. This paper deals with the landing planning phase, in which the spacecraft uses imagery to precisely position itself with regard to the asteroid, and choose a landing site.

Spacecrafts have, at any time, knowledge of their own position which is accurate enough for most parts of a mission. However, when it comes to the landing planning phase, it is necessary to have a very accurate estimation of the position in order to avoid uneven or hilly sites corresponding to the craters. We describe here a visual positioning system for spacecraft during a landing phase on asteroid. The position obtained by this system is expressed in term of asteroid latitude and longitude.

We first give an overview of the system in Section 2. In Section 3 we propose a method to obtain an ellipse representation of the craters present in asteroid images. Section 4, describes a pose estimation algorithm using ellipse matching.

2 Overview

We assume that the approximate position of the spacecraft, as well as the camera parameters, are known within some precision error and that a 3-D model of the asteroid is available. This model, which can be built during the orbiting phase, includes the position and the geometry of most of the craters present on the asteroid surface. The system described in this paper consists of two separated modules. First, the craters are extracted from the 2-D image and represented as ellipses. Then, in order to find the best transformation between the image and the previously generated 3-D model, the ellipses are matched with those contained in the model. Figure 1 gives a top level overview of the system. The estimation of the spacecraft position is related to the problem of 3-D object recognition from 2-D intensity images which can be addressed using matching models. Since craters constitute excellent "interest features" on asteroids, the 3-D model database contains a description of the asteroid craters described as ellipses.

We describe, in Section 3, a method to infer the curvature from noisy sparse data. This method, based on the tensor voting approach, is used to extract ellipses from edge images. The tangent directions of the input data are obtained from edge images, then the curvature is estimated in order to obtain the orientation of the normals. Finally this information is used to generate a dense saliency map describing elliptical shapes. In order to extract craters of different sizes, this process is performed at several scales.

Once the craters are extracted, a matching of the scene with the 3-D model is achieved using these features. Since we have an approximate knowledge of spacecraft position, there is no need to use *all* the craters of the 3-D model to perform this process. Only the craters present on the side of the asteroid viewed by the spacecraft will be considered. Section 4 describes an algorithm which consists of tests of the rigid transformations between the model projection and the scene by evaluating the matches between the ellipses corresponding to the craters. The

evaluation is a vote process based on the position and the parameters of the ellipses. The best transformation obtained using this method gives us the position of the spacecraft in term of asteroid latitude and longitude.

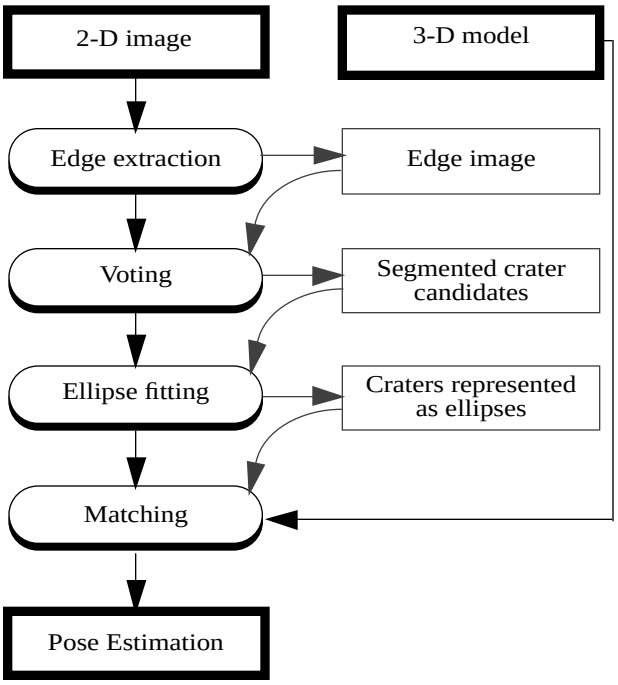

Figure 1. Overview of the System

The system has been tested on images of Phobos, one of the two Mars moons, using an analytical 3-D model described in [3]. Moreover, a synthetic model has also been used to test the crater detection process.

3 Crater extraction

Craters, on asteroids, are the result of collisions that create a bowl-shaped depression around the impact point. In an image, these craters should produce shapes well approximated by ellipses. In fact, models, manually generated by JPL, such as the model of Phobos (see Figure 2) described in [3], consist of the sum of two separate models. The first one describes the surface of the asteroid with spherical harmonic expansion terms. It gives the distance from the center of the asteroid to the surface in function of the latitude and longitude parameters. The second model, added to the surface model, corresponds to the craters and is based on an elliptical description of the craters:

$$c(k, D, R) = \begin{cases} -kD\cos\left(\frac{R}{D}\pi\right) & \text{if} \quad 0 \le 2R \le D \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

where k is the crater depth-to-diameter ratio, D is the diameter of the crater, and R is the distance from the center of the crater. Note that the projection of the crater representation on the plane tangent to the surface is a circle.

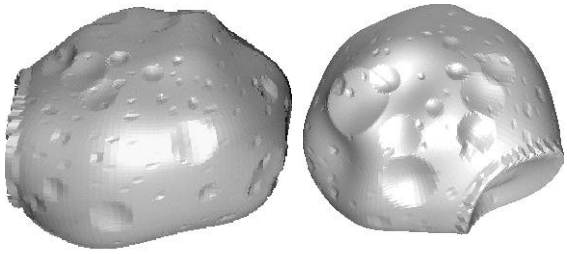

Figure 2. Two views of the 3-D model of Phobos

Among existing object recognition methods (see [4] for an extensive review), the Hough transform constitutes a popular approach for geometric shape detection. The principal concept of this method is to compute a mapping between a parameter space that defines the shape to be detected and an image space. The computational and memory cost of this method increase exponentially with the number of parameters.

Extensions of the Hough transform method as described in [1] and [6] have permitted to reduce the cost of this method, when dealing with ellipse extraction, by changing the parameter space decomposition. A side effect of these extensions is a lack of robustness, especially when the shape of the object is not a perfect ellipse and the data is very sparse.

Another drawback of the methods using Hough transform is that the grouping and the parameter computation are performed separately. Thus, while these methods could be easily used to detect a single ellipse or several ellipses having a similar geometry, the detection of multiple ellipses having different geometry cannot be handled easily.

We propose a method which can perform this task in a single pass by doing grouping and extraction simultaneously. Our method is based on an approach presented in [5] and extended in [8] and [10] which permits to infer curves and junctions from sparse data as well as surfaces in the 3D case. This approach uses tensors as representa-

tion and the voting technique in order to propagate the information.

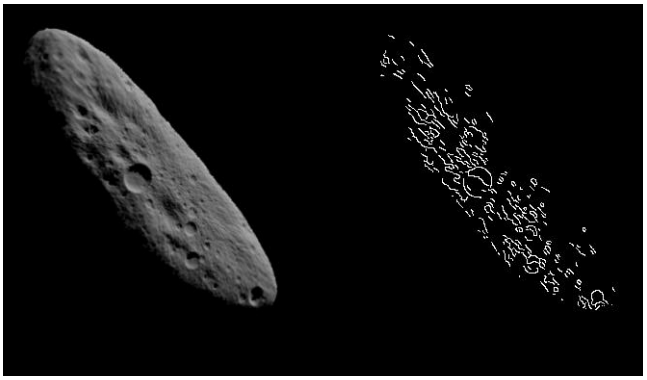

Figure 3. Synthetic model projection and edge image

The edges are extracted by applying to the data an edge detection model described in [2]. The tangent direction associated to each edge point is computed using the tensor voting technique. The curvature at each input site (the edge points in our case) is estimated. To this end, we describe a method that can infer curvature estimation from noisy sparse data. This information is then processed in order to produce a dense saliency map which models curves and outlier contained in the input data. The connected component of the local saliency maxima are then labeled as being part of a same crater. Finally, an ellipse fitting is applied to the local maxima of the saliency map. This algorithm, summarized in Figure 4, is performed at different scales.

3.1 Tensor voting

In 2D, an edge point can be classified as belonging to a curve, or being an outlier. These two classes differ in term of direction: while the curve point has a well defined direction, the outlier is characterized by a maximal uncertainty of direction. Therefore, discrimination between curve point and outlier point can be achieved by comparing direction (un)certainty.

Tensor representation. Direction and uncertainty can be formalized using a second order symmetric tensor derived from the covariance matrix T that describes the distribution of curve directions.

Using the eigenvalues λ_1, λ_2 of T $(\lambda_1 > \lambda_2)$ and their associated eigenvectors U_1, U_2, T can be expressed as:

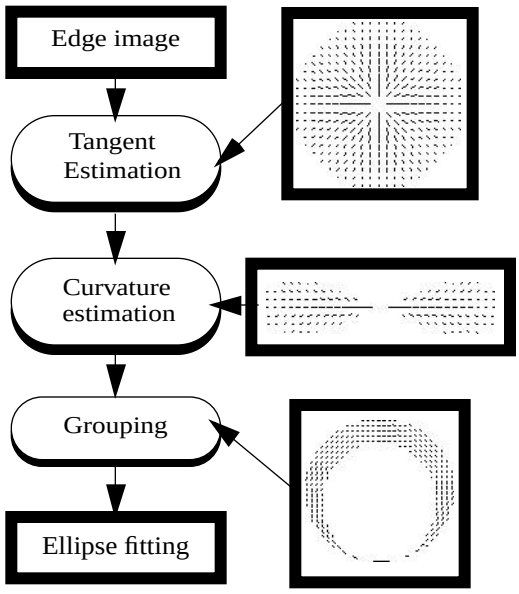

Figure 4. Description of the craters extraction at a given scale

$$T = (\lambda_1 - \lambda_2)S + \lambda_2 P : \qquad (2)$$

where $S = \hat{U}_1\hat{U}_1^T$ and $P = \hat{U}_1\hat{U}_1^T + \hat{U}_2\hat{U}_2^T$. In the mainframe of the described approach, $(\lambda_1 - \lambda_2)$ corresponds to *saliency* and λ_2 to the uncertainty of the direction defined by $\hat{U}_1\hat{U}_1^T$.

Voting process. The aim of the voting process is first to generate a tensor description of the input data sites and then to densify this information in order to obtain a tensor representation at each pixel location of the input.

The direction of a site as well as its associated certainty can be derived from the position, direction and (un)certainty of the site neighbors. Therefore, a vote is performed locally on each site of the input data in a convolution-like operation using two predefined tensor fields depicted in Figure 5-a and Figure 5-b. The result of this operation is a tensor.

The point tensor field (see Figure 5-a) is an isotropic field having strength modelized by a Gaussian function. The stick tensor field (see Figure 5-b) privileges a direction and reflects the choice of favoring curves of small and constant curvature. The strength at a given site is a Gaussian function of two parameters: the distance from the origin and the curvature of the circle going through the origin of the tensor field and the site. Figure 5-c represents a cir-

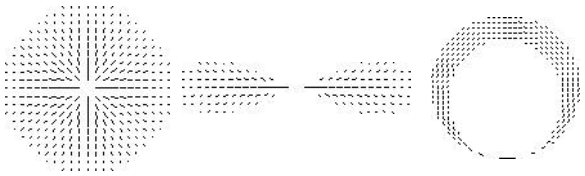

Figure 5. Voting tensor fields: (a) point field, (b) stick field and (c) circle field

cular field that will be used to obtain a dense saliency map of the circular features.

Direction inference. Prior to any computation, each input data site is encoding as a point tensor, since no direction should be privileged. The vote is then cast between sites belonging to input data which corresponds, in our application, to the edge image. As a result, the tangent direction at each edge point is obtained.

3.2 Curvature inference

The edges of a craters are generally represented in a contour image as three separate curves having a tangent direction normal to the direction of the light (see Figure 3). The middle curve corresponds to the limit of the shading and the others to the boundaries. In order to group the edges belonging to the same crater, it is useful to have a knowledge of the inside and outside of the crater and therefore to compute the *oriented* normal on each site of the input.

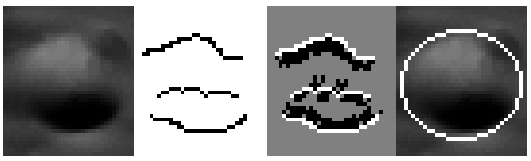

Figure 6. From left to right: a typical crater, associated edges, curvatures estimation and ellipse fitting result.

Curvature inference. As described above, the definition of the stick tensor reflects the choice of favoring curves of small and constant curvature. Therefore, taking into account the nature of the curves which are favored by the tensor voting, we propose a new tensor field specifically designed for the estimation of the *curvature*. The curvature tensor field based on the stick field, is made of components containing not only the direction and the associated strength, but also the curvature corresponding to the circle going through the origin of the field and the considered component.

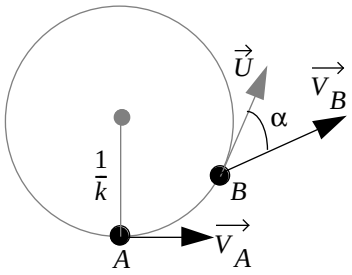

Figure 7. Voting for curvature

Let us consider the vote cast by a site A for another site B while using the curvature tensor field as voting support (see Figure 7). The result of this vote relies on the curvature of the circle defined by the direction of the site A and the location of the sites A and B. Thus, the weight associated to this vote is:

$$w = G(d_{AB})H(\overrightarrow{V_B} \bullet \vec{U}) \tag{3}$$

Where $G(d_{AB})$ corresponds to a Gaussian function of the distance between AB and H is a sharp Gaussian-like function ($\sin(2x)^2$ for instance). This weight ensures us that the nearest neighbors will be the most influent. Moreover, sites having incompatible tangent direction will give a vote of lesser importance.

The vote is stored by each input site in a distribution form defined by:

$$f(k) = \sum_i w_k^i \tag{4}$$

Where $\{w_k^i\}$ are the weights associated to the vote casted by the neighborhood for the curvature k. By computing the mean and the standard deviation of this distribution we obtain the curvature and the uncertainty.

It should be noted that the described method gives an estimation of the curvature which is generally under estimated since we are favoring curves of small and constant curvature. However, in the mainframe of our application, we are only interested in the *sign* of curvature in order to obtain the orientation of the normals.

3.3 Ellipse fitting

Having computed the tensor description of each input data, it is possible to infer circular features by computing a

dense saliency map. This map is generated by casting votes between any location, should it be part of the input data or not. To this end, we use a circular field depicted in Figure 5-c. This field is made of circles intersecting the origin and tangent to the axe Ox. The strength associated to each site is a Gaussian function of the distance. When applied to the data, this field propagates the proprieties of circular forms among their neighbors. As a result, we obtain a dense saliency map whose maxima are the craters boundaries.

By applying a hysteresis thresholding to this map the edges points belonging to each connected region of saliency extrema are labeled as being part of a same crater. In order to obtain a compact and even more accurate representation of the detected craters, the labeled edges representing the craters are processed using an ellipse fitting method proposed in [9]. This robust method is ellipse specific and use a non iterative least square algorithm.

3.4 The scale problem

A well known problem with using grouping methods on sparse data is the scale dependence of the process. Depending of the size of the field supports, the inferred direction and the saliency of the data will differ. As a result, a small field will only detect small craters while a bigger one will tend to merge several neighbor craters into one. Since we are processing images obtained from an orbit position, the distance between the asteroid and the spacecraft does not change significantly. Hence, there is no reason to consider a large spectrum of scales. In all the imagery processed so far and with very few odd exceptions, the craters vary by a factor 4 in diameter. We therefore repeat the voting process for the three different scales using support of 8, 16 and 32 pixels for our predefined field.

Figure 8 shows the craters detected at different scales on the image displayed in Figure 3 which has been generated using a synthetic model of an asteroid. Using as input, the edge image of Figure 3, the algorithm is able to detect approximately 30 ellipses. 5 of them do not correspond to a crater. The process performed on a 300 by 300 image takes less than 60 seconds on a SGI O2.

4 Pose estimation

The position of the spacecraft is achieved by performing a matching of the detected craters with the craters of the 3D model. Note that in the landing plan phase, only the latitude and longitude of the spacecraft in the asteroid coordinate system are necessary, we do not need to refine

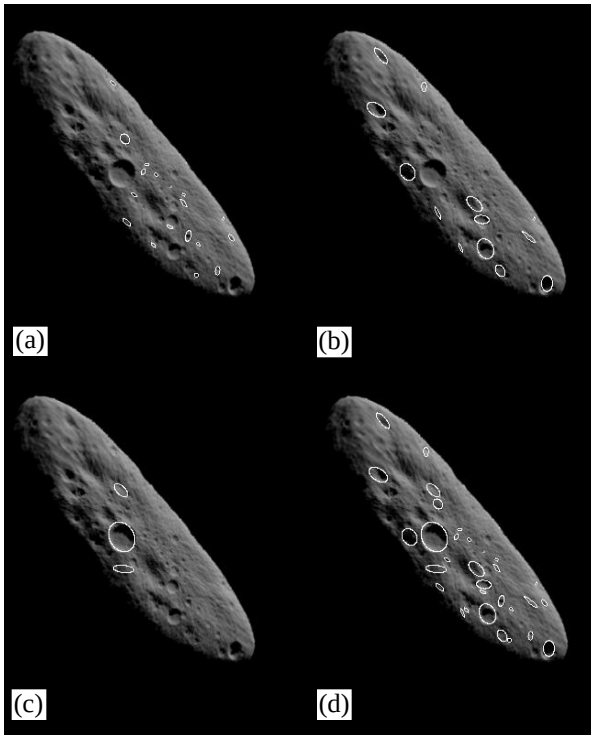

Figure 8. Result of crater detection on the synthetic model. (a) to (c) results at 3 different scales. (d) final result

the estimate of height of the spacecraft with respect to the asteroid center.

The method used to achieve this matching is based on a model-based object recognition approach described in [7]. This approach use an affine invariant representation of the data and the representation of the model is stored in a hash table. The algorithm consists of two major steps. First the model data are preprocessed in order to obtain affine invariant representations which are stored in an hash table. Then, during the matching process, several representations of the input data are compared to the representation of the hash table using a vote process. When a satisfactory score is obtained, the transformation between the model representation and the scene is tested using two successive verification steps. The first one uses a least squares match and the second one compares specific properties of the data. If one of these verification fails, then an other representation is tested.

4.1 Hypotheses generation

One of the constraints of our application comes from a limited memory size available on the spacecraft computer.Therefore our method does not use a hash table.

While this choice reduces significantly the memory usage, it has the important drawback of increasing the complexity of the algorithm.

Another specificity of our application is the approximate knowledge of the spacecraft position and the camera parameters. It is then possible to generate, with some error, a projection of the 3D model corresponding to the side of the asteroid viewed by the spacecraft. Therefore we can restrict the matching to a 2D problem. By doing so, the complexity of the process is decreased noticeably and we can expect a better robustness.

Commonly, matching algorithms consist in finding the best transformation between the model data and the scene data. Since we are principally interested in processing orbit images, the depth of the asteroid surface is negligible compared to the distance from the spacecraft to the asteroid.

Therefore, we restrict the space of the transformations to the 2D rigid transformation (translation, rotation and scale). Note that by doing so, the complexity of our algorithm is once more reduced.

4.2 Matching

Our matching algorithm evaluates the transformation using a two-step examination. First the pairs defining a transformation candidate are tested in order to evaluate if the candidate is valid. Then, the transformation is evaluated by comparing the transformed parameters of the scene ellipses with those of the model projection. If the number of good matching is considered sufficient, the process stops, otherwise, another transformation is evaluated.

We have tested the matching process on Viking images of Phobos. Figure 9-a shows the detected craters while Figure 9-b depicts the position of the craters present on the projection. Using the result of the matching, the image can be superimposed on the model projection as in Figure 9-c where detected craters are in white and model craters in black.

Depending on the resolution of these images, 10 to 30 craters are detected using the crater extraction method described in Figure 3. The 3-D model, as described in [3], is made of nearly 300 craters which means that a projection of Phobos can contain approximately 150 craters. Since it would require too much computation time to perform a matching between 30 craters of the scenes and 150 craters of the model, we restrict the number of the scene craters to the 10 craters corresponding to the most important saliency (i.e.: certainty). In the worst case, when all the possible transformations are evaluated, the process takes less than 30 seconds. on a SGI O2.

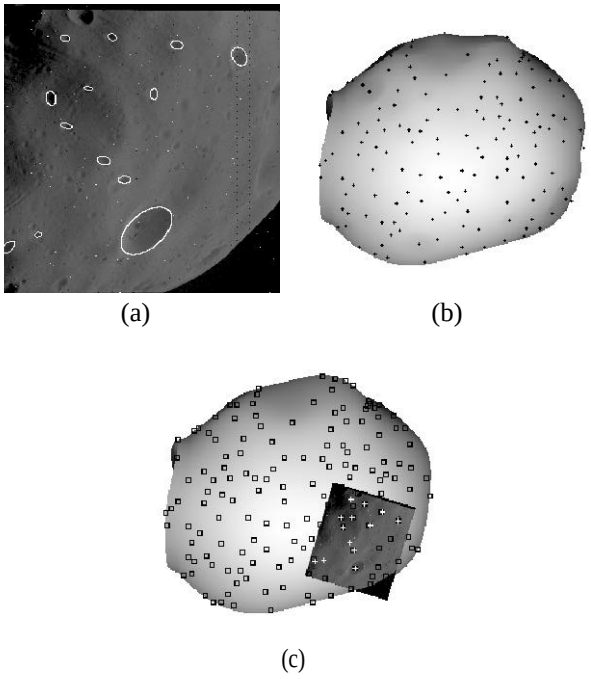

Figure 9. Matching results: (a) detected craters, (b) 3D model craters, (c) superimposition of the image on the model projection

Using the position of the image and the camera parameters, we easily obtain the actual position of the spacecraft in term of latitude and longitude in the asteroid coordinate system.

5 Conclusion

We have presented a visual positioning system for spacecraft for landing plan phase on asteroids. It is based on two principal modules. First, an ellipse extraction method performed at different scales permits to detect the craters present on the asteroid image. Using tensor voting, techniques, the tangent directions are obtained from edge images. The oriented normal at each edge point are then computed. To this end, we have proposed a new robust method to infer the curvature from noisy sparse data. The oriented normals are then used to infer saliency map of circular features present in the image. Finally, the most salient craters of the image are then compared to the 3-D model in order to obtain an accurate position of the spacecraft. Using a rough estimation of the spacecraft position, a projection of the 3-D model is computed. A matching is

then achieved in order to find the best transformation between the detected craters and the craters present on the projection.

Future work should proceed along two axes. Since the system is planned to be used by JPL during a NASA mission, further tests will be performed on images of other asteroids. Another issue will be related to the scale problem. While we have proposed an approach to extract ellipse features at different scales, a true multiscale analysis for tensor voting is under investigation.

References

[1] A.C. Aguado, M.E.Montiel and M.S. Nixon, "Arbitrary shape Hough Transform by Invariant Geometric Features", *proc. IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, Florida, 1997.

[2] R. Deriche, "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector", *International Journal of Computer Vision*, vol. 1 no. 2, pp 167-187, 1987.

[3] T.C. Duxbury, "An Analytic Model For The Phobos Surface", *Planet Space Sci.*, Vol. 39, No 1/2, pp. 355-376, 1991.

[4] W.E.L. Grimson, "Object Recognition by Computer: The Role of Geometric Constraints", *MIT Press*, 1990

[5] G. Guy and G. Medioni, "Inference of Surfaces, 3D Curves and Junctions from Sparse, Noisy 3D Data", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, pp. 1,265-1,277, 1997.

[6] C.Ho, L. Chan, "A Fast Ellipse/Circle detector using geometric symmetry", *Pattern Recognition*, vol. 28, pp. 117-124, 1995.

[7] Y. Lamdam, J.T. Schwartz and H.J. Wolfson, "Affine Invariant Model-Based Object Recognition", *IEEE Trans. Robotics and Automation*, vol. 6, no 5, October 1990.

[8] M. S. L. Lee and G. Medioni, "Inferring Segmented Description from Stere Data", *proc. Computer Vision and Pattern Recognition 1998*, pp. 346-352, Santa Barbara, California, June, 1998.

[9] M. Pilu, A. Fitzgibbon, R.Fisher, "Ellipse-specific Direct least-square Fitting", *IEEE International Conference on Image Processing*, Lausanne, September 1996.

[10] C. K. Tang and G. Medioni, "Inference of Integrated Surface, Curve and Junction Descriptions from Sparse 3D Data", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1,206-1,223, 1998.